A METHOD TO PREDICT THE RELIABILITY OF MILITARY GROUND VEHICLES USING HIGH PERFORMANCE COMPUTING

David A. Lamb*, David Gorsich, Dmitriy Krayterman
U.S. Army RDECOM-TARDEC
Warren, MI 48397

K.K. Choi, Ed Hardee
University of Iowa
Coralville, IA 52242

Byeng D. Youn
Michigan Technological University
Houghton, MI 49931

Dan Ghiocel
Ghiocel Predictive Technologies, Inc.
Pittsford, NY 14534

ABSTRACT

A method is presented to use the massively parallel environment of High Performance Computing (HPC) to more rapidly compute the reliability prediction of military ground vehicles. Current work, and future plans are discussed. Challenges already surmounted are indicated, as are those still to be met.

1. INTRODUCTION

A major challenge to current military operations is the lack of a rapid and accurate method to assess ground vehicle reliability using modeling and simulation. Reliability is a highly complex field, involving many different physics-of-failure, including fatigue, thermal stress, corrosion, and erosion. Reliability also involves uncertainty in the input data, and is ultimately stated as a probability. In fact, stochastic methods, rather than deterministic, characterize this field. The assessment of the reliability of a complex mechanical system in many different physics-of-failure is a huge computational challenge.

The Army wants to improve the reliability of its ground fleet, and to do that requires an accurate assessment of the reliability of a design using modeling and simulation. Currently, such analyses take a large amount of computer time and are not able to deliver results in a rapid manner, consistent with the needs of the decision making process. This must be addressed, to satisfy the need to design for better reliability.

To impact the decision making for ground vehicles, we are using High Performance Computing (HPC) to speed up the time for analyzing the reliability of a design in modeling and simulation. We use parallelization to get accurate results in days rather than months. We can obtain accurate reliability prediction with modeling and simulation, using uncertainties and multiple physics-of-failure, but by utilizing parallel computing we get results in much less time than conventional analysis techniques.

1.1 The Scope of the Problem

Prof. K.K. Choi, of the University of Iowa, performed an optimization of the design for an A-arm on a military ground vehicle (a Stryker), using no sources of uncertainty and only one physics-of-failure. This was not done in any parallel way. He reported using 768 FEA runs of small-sized models (30K – 200K DOF) and taking 3.55 days of compute cycles. This was just for a single component and a single physics. He estimated that to do a full vehicle would take at least 100 times that, or 76,800 FEA runs and 355 days in serial mode. But, he reports, the FEA are all largely independent and could be done in parallel. Utilizing 1,000 processors each capable of doing a single FEA run on a small-size model in serial, he projects that the turn-around time drops to below half a day.

1.2 Our Goal

We are planning for something even more ambitious, using four or five physics and many sources of uncertainty requiring Monte-Carlo techniques. Estimates climb into the tens of millions of FEA runs of small-sized models,

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 NOV 2006	**N/A**	**-**

4. TITLE AND SUBTITLE	
A Method To Predict The Reliability Of Military Ground Vehicles Using High Performance Computing	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **U.S. Army RDECOM-TARDEC Warren, MI 48397**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002075., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**6**	

and hundreds of years of clock time if done in serial. Fortunately, there is no need to do this in serial, since most of the FE analyses are independent, and we can parallelize. Utilizing 10,000 processors to parallelize the FEA runs will keep the turn-around time below two weeks. To be useful in influencing the acquisition process, turn-around times longer than week are not helpful. Unfortunately, we cannot immediately jump to using 10,000 processors, but will have start out more modestly and grow to that level.

2. THE METHOD

Some key features of this method are that it is physics-based, starting from first principles, rather than heuristic, and it seeks to handle interactions between different components of the ground vehicle and different physics-of-failure on this basis (non-heuristic). We are seeking methods to compute fatigue, thermal stress, corrosion and other causes of failure using physics-based equations as can be found in textbooks or handbooks, not simply by a heuristically generated response surface or some other 'rule of thumb' based more on statistical manipulation than physics first principles. We want to predict the reliability of the ground vehicle starting at the material level, working up through components, assemblies and subsystems to the system level, and have a good scientific basis (rather than just a statistical basis) for each step.

Understandably, this takes a massive amount of computing to accomplish. We parallelize at several different levels, including putting different components onto separate sets of processors and putting different physics-of-failure onto their own processors. With a scheme of dividing the problem up by parts of the vehicle, failure modes, and dealing the stochastic uncertainty using multiple processors, we are relying on the High Performance Computers to make this solution run.

The intended end use of this method is to quickly and accurately generate a prediction of the reliability for a proposed design, so that this prediction can be used for trade-off studies or for optimization of the design. As such, the method must only use input which would be generally available during the design cycle when trade-off studies are made. Also, to actually have any influence on the final design, the prediction must be accomplished in a short amount of time, so the results are available for the next design iteration. We expect that unless a prediction can be made in a week, we will miss the opportunity to guide the design loop process toward greater reliability.

2.1 Massive Number of FEA Runs

The main idea that we are using is that the reliability analysis incorporates a large number of FEA analyses, most of which are independent. The greatest speedup in time to final answer will come from spreading the FEA runs across a large number of processors to be executed in parallel. This will require methods to break the large scale systems into lower scale ones, and methods to break apart different physics-of-failure into separate analyses loosely coupled with each other. Also, an automated process for generating the necessary multiplicity for the Monte-Carlo technique to deal with the uncertainties will be needed. Finally, methods to consolidate results back up the system level, to generate the report, will be required.

2.2 Course Grain versus Fine Grain Parallelization

We did a preliminary study to decide if we should try to parallelize a single FEA run, or just run lots of FEA runs (each in serial) simultaneously. The results of this study showed that our typical FEA runs are not particularly large, but we need a lot of them run. Culling from an analysis of a Stryker A-arm done using only a single physics-of-failure (fatigue) here are some results.

- For the Stryker A-arm a typical 4 iteration deterministic optimization takes about 3.55 days. That includes 768 FE analyses [$768 = 4$ iterations $\times$ 24 load cases $\times$ (2 function evaluations per iteration + 6 derivatives for sensitivity analysis)].
- Thus 100 runs for a Monte-Carlo analysis may very well require $3.55 \times 100 = 355$ days.
- This is just in durability, without considering other physics-of-failure, we are involved with a VERY large number of FE analyses ($768 \times 100 = 76{,}800$ analyses) of SMALL size FE models (30~200k DOF).
- This is only one component (the A-arm) in a vehicle with hundreds of components to be analyzed.
- A full vehicle (100 components) with four physics-of-failure and 100 Monte-Carlo points for generating the distribution should take 3.55 days x 100 x 100 x 4 $\approx$ 389 years.
- But the same analysis will consume 768 x 100 x 100 x 4 = 30,720,000 FE analyses, each in the 30~200k DOF range.

See figure 1 for an example of this.

Thus, speed up could be achieved significantly more by carrying out a number of FE analyses simultaneously, rather than trying to make each FE analysis faster. Parallelizing by putting one FEA on each processor but running 1000 at a time counts more than spreading a 200k DOF FEA across 100 processors.

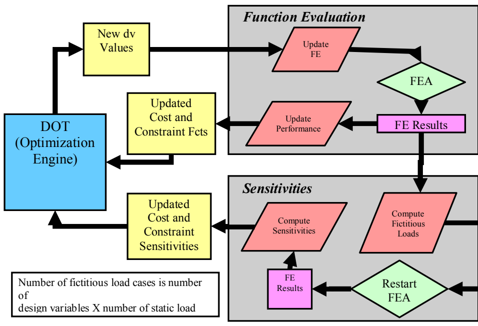

Figure 1. Example of method described.

As it turns out, while this is a very good way to parallelize the method, it leads to a significant challenge for the project, as we will discuss later in this paper. Right now, it is not clear how to solve this problem without help from software vendors.

2.3 The Challenges

We expected to find several challenges in the computational process caused by the need to generate, coordinate, and finally consolidate the runs on lower scales. At the lowest level, we plan to rely on native scheduling/queueing software to coordinate putting the many FEA runs onto the processors.

We did find a number of challenges. We were unable to purchase the work flow software we wanted due to a budget limitation, so we had to script our own work flow control. This provided a challenge.

We also encountered a challenge obtaining the base data needed for the study, particularly in the area of uncertainty distributions for the material properties of the steel in the part being studied. This is discussed further below.

However, it turned out that the largest challenge we encountered was actually budgetary, but tied in with the licensing policy of some software we planned to use. This is discussed more fully below, and will clearly impact any future work done along these lines.

3. THE PROJECT

We made the runs in September-October 2006 on the High Performance Computers located at U.S. Army RDECOM-TARDEC in Warren, MI. We describe here the results seen in these runs.

We analyzed the lower driver's side A-arm from the M-1097 HMMWV. (See figure 2.) This was analyzed to improve the design for fatigue life. We chose this part because it was very similar to another study done using serial processing earlier, and there was thought to be a lot of data available for this vehicle and this part.

We wanted to do a multi-scale, multi-physics analysis of a subsystem, but as the saying goes, you have to walk before you can run. We were limited on resources we could bring to the pilot project and found that the only way to get anything run with the limitation on our resources was to be more modest in our immediate goals. This caused us to restrict ourselves for the pilot project. We only did a single component and a single physics-of-failure.

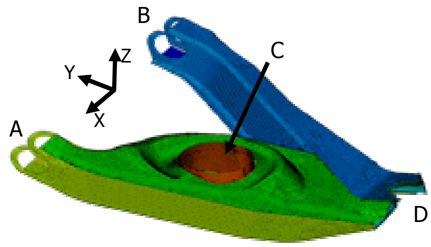

Figure 2. HMMWV lower A-arm.

3.1 The Computer Hardware

Three computer systems were used for this project. The first was an SGI Altix 3000 with 8 1.3 GHz Itanium 2 processors, 8 Gbytes memory and 72 Gbytes local disk space. The second was an SGI Origin 3900 with 24 MIPS R16000 processors, 24 bytes memory and 72 Gbytes local disk space. The third was an SGI Onyx 350 with 32 MIPS R16000 processors, 32 Gbytes memory and 36 Gbytes local disk space. All three are located in Warren, MI at eh Detroit Arsenal, and are part of the DoD HPC Modernization Program.

3.2 The Operating System Setup

The operating systems used were SGI's propriety version of UNIX known as IRIX (used on the Onyx and Origin machines) and LINUX (used on the Altix machine). All systems ran LSF for the queueing system.

3.3 Reliability/Fatigue Analysis software

We used several pieces of propriety code from the University of Iowa for this project. These included a fatigue analysis software called DRAW, a design sensitivity software called DSO and a reliability-based design optimization software, called RBDO. All three were ported from the University of Iowa to TARDEC's HPC center and installed for run. (See figure 3.)

In addition to these, we made use of some numerical analysis software called DOT from

Vanderplaats. This was used primarily to perform the optimization in the loop.

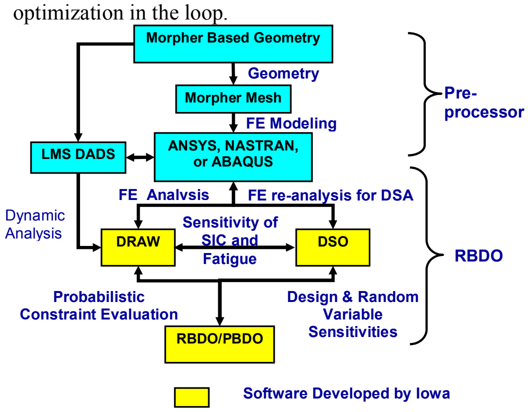

Figure 3. Software loop diagram.

3.4 Finite Element Analysis solver

We needed extensive use of a finite element analysis solver. For this, we choose to use NASTRAN from MSC. This turns out to be a significant roadblock and challenge for projects of this type. To accomplish significant parallelization of the method, we required that multiple copies of an FEA solver be running on different processors, solving variations of the same analysis, in parallel. Unfortunately, we found that most vendors of FEA code treat this situation as requiring a license for each solver we run. So, to run on sixteen processors required having sixteen licenses, and to run on a hundred processors would have required a hundred licenses.

So we find that this becomes a very costly hurdle for expanding this project. We are not likely to make the progress we want, if we must purchase several hundred licenses for an FEA solver to parallelize across hundreds of processors. A better way of handling this must be found to facilitate further progress.

For our pilot project, we negotiated with MSC to obtain a limited time window where we could use sixteen NASTRAN licenses for this project, but only on an experimental basis to demonstrate the method we are developing. We will then need to start buying licenses for future work.

It will be very advantageous for future work in this area to find a vendor of FEA software that will offer a better pricing scheme. What would seem best would be for the vendor to allow for multiple (hundreds?) runs of their software to be made in parallel, across hundreds of processors, on variations of the same problem, for some fixed price. Perhaps some control could be imposed to insure that all the runs are variations of the same base problem, as a way to prevent fraud. While it is not clear how to adequately protect the software vendor's interest while keeping costs reasonable, still it is obvious that without something like this, the potential for this method is very limited. We cannot easily see how to expand the current method to a hundred or more processors if we must effectively buy a license for the FEA solver for each processor utilized.

3.5 Parallelization and work flow control

RBDO demands multiple reliability analyses at a given design. In the pilot study, refined reliability analyses for n number of the active/violate probabilistic constraints are planned to be executed in a parallel manner on HPC, as shown in Fig. 4. Thus, only several processors are needed to parallelize the entire process of reliability analysis. Up to now, the parallelization has been successfully tested using Load Sharing Facility (LSF) on Linux Cluster (10-processor/5-node) at Michigan Technological University (MTU).

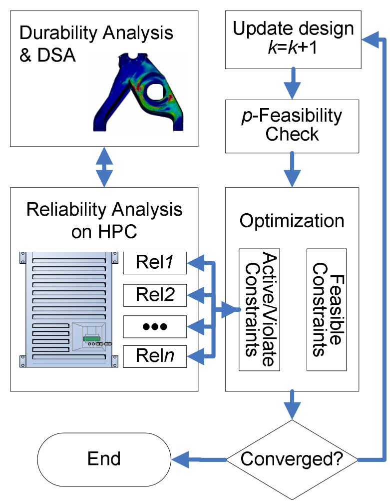

Figure 4.Parallelization of Reliability Analysis Using LSF.

3.6 Preprocessing software

We required multibody dynamic analysis of the whole vehicle to obtain loads for the fatigue analysis. This dynamic analysis was done once, in a preprocessor step, using the DADS software from CADSi (now part of LMS). This was not done during the parallelization stage, and the same loads were used throughout the entire pilot run. The DADS software was just for preprocessing the dynamics laods.

We also used Hypermesh for creating the original mesh on the part we were analyzing. This was done once in a preprocessor step. NASTRAN was run in a preprocessor step to determine 'hot spots' and pre-

configure the fatigue solving step. (See figure 5.) This required only a single NASTRAN license, as this run was made prior to any parallelization of the method.

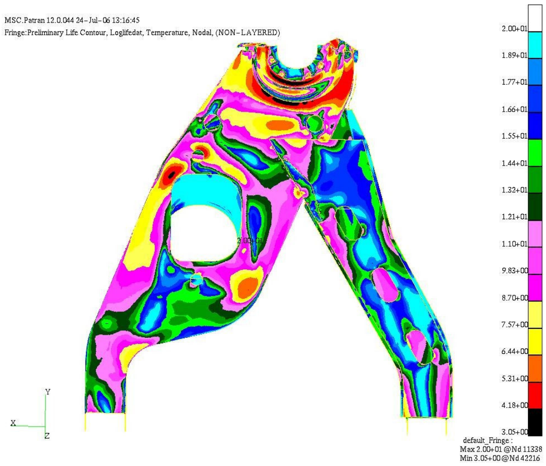

Figure 5. Preprocessor hot spots on A-arm.

3.7 Problem Definition of Design and Random Parameters

The A-Arm is composed of 20 pieces of plate including three small reinforcements, which are made of High Strength Low-Alloy (HSLA) SAE 950X Steel. Among the plates, seven plates are controllable: upper and lower main arms, upper and lower support arms, and three reinforcement plates. They are defined as design and random parameters. In addition, five fatigue material properties are considered as random parameters [Socie 2005]. Table 1 summarizes both design and random parameters.

Table 1. Properties of Design and Random Properties

Random		d_L	d, μ	d_U	Dist.	σ
Controllable	X_1	0.100	0.120	0.500	Norm	0.012
	X_2	0.100	0.120	0.500	Norm	0.012
	X_3	0.100	0.180	0.500	Norm	0.018
	X_4	0.100	0.135	0.500	Norm	0.0135
	X_5	0.100	0.250	0.500	Norm	0.025
	X_6	0.100	0.180	0.500	Norm	0.018
	X_7	0.100	0.135	0.500	Norm	0.0135
Noise	X_8		802		LogN	96.24
	X_9		0.26		LogN	0.1092
	X_{10}	N.A.	–0.09	N.A.	Norm	0.0225
	X_{11}		–0.62		Norm	0.1426
	X_{12}		205		LogN	20.50

In Table 1, $X_8 = \sigma'_f$ is the fatigue strength coefficient; $X_9 = \varepsilon'_f$ is the fatigue ductility coefficient; $X_{10} = b$ is the fatigue strength exponent; $X_{11} = c$ is the fatigue ductility exponent; $X_{11} = E$ is the modulus of elasticity.

Ten percentile (10%) coefficient of variation (COV) is used to model for geometric random design parameters and the modulus of elasticity.

4. THE PAYOFF

When talking about reliability, it is important to consider 'total lifecycle cost' as the relevant measure. This is because adding reliability often costs extra at the front end (during research, development, design and manufacturing) but realizes savings during the Operations and Sustainment phase of the life cycle due to reduced costs to keep the vehicle available. To understand the value added by the increased reliability, the key is to balance the added up front costs against the savings later on, in other words, to look at total cost across the entire life cycle of the vehicle.

Also, the projected savings from improved reliability is often based on the current level of reliability we start with (based on the law of diminishing returns). If a fleet is showing low reliability before efforts begin, then a large cost savings due to improved reliability is possible, but it is hard to realize great savings when starting from a fleet of very reliable vehicles. Based on current data from Army fleets, it appears that improved reliability in Army ground vehicles has a potential for very respectable cost savings.

Total savings will also be a function of the number of similar vehicles in the fleet based on the improved design. It is obviously easier to realize large cost savings from improving the reliability of a design with 10,000 fielded vehicles that improving the design that only fields 50 vehicles. Still, once methods are developed to improve the reliability of a design, and the cost to develop the methods is recouped from improving the design of a few vehicles, the same methods will still be available to use on all other vehicle designs with little added cost. The key, therefore, is to apply the new methods to a few systems where the development costs of the new methods can be quickly recouped, and then deliver to the Army a 'paid for' tool to improve the reliability for other platforms.

It is reasonable to assume that tens of millions of dollars in total life cycle cost savings might be realized for a fleet of a single ground vehicle design due to improved reliability designed in from the beginning. (Savings will be spread across the whole life cycle and across the fleet of similar vehicles.) If this method can be used to improve the design of just ten future vehicles, with various sizes of fleets and various results of reliability improvement for each, the method could potentially lead to savings of hundreds of millions or even billions of dollars. Even just

one vehicle design will more than repay the costs of developing and implementing the method, based on modest reliability improvements to the design from the use of this tool.

CONCLUSIONS

While the Army struggles with the reliability of its current and future fleets of ground vehicles, there is a great need for a tool of this sort. We want to make it a good tool, one based on physics and not heuristics, and one that considers system level reliability with interactions between components and between failure modes captured. This requires the massively parallel environment of High Performance Computing to be realized quickly enough to impact the design loop. We are working to build this technique, make it multi-physics and multi-scale and non-heuristic. As this project progresses, we will add additional complexity to the models and generate predictions that encompass more of the true range that reliability should include.

The most significant hurdle still to be made is how to obtain, at a reasonable cost, sufficient licenses for FEA solving software to parallelize across hundreds of processors as desired.

ACKNOWLEDGMENTS

We would like to acknowledge the support of all our institutions. We also want to acknowledge the kind donation of software from MSC Software Corporation and Vanderplaats Research and Development, Inc. to this effort.

REFERENCES

Davis, TP., 2004: "Science engineering and statistics"; Henry Ford technical Fellow for Quality Engineering; Ford Motor Co.; http://www.timdavis.co.uk/research.htm.

Kokkolaras, M., Mourelatos, Z. P., and Papalambros, P. Y., 2004: "Design Optimization of Hierarchically Decomposed Multilevel System under Uncertainty", Proceedings of the ASME 2004 Design Engineering Technical Conferences, Salt Lake City, Utah, September 28 - October 2, 2004, DETC2004/DAC-57357.

Hartmanis, Juris, Stearns, Richard, 1965: "On the computational complexity of algorithms", Trans. Amer. Math. Soc. 117 (1965), 285-306.

Youn, B.D. and Choi, K.K., 2004: "Selecting Probabilistic Approaches for Reliability-Based Design Optimization," AIAA Journal, Vol. 42, No. 1, 2004, pp. 124-131.

Choi, K.K. and Youn, B.D., 2001: "Hybrid Analysis Method for Reliability-Based Design Optimization," 27th ASME Design Automation Conference, September 9-12, 2001, Pittsburgh, PA.

Socie, Darrell F., 2005: Probabilistic Aspects of Fatigue, *http://www.fatiguecalculator.com/pdfs/ProbabilisticFatigue.pdf#search=%22SAE%20950X%20deviation%22*